



TECHNISCHE
UNIVERSITÄT
DARMSTADT

ULB

The PRORETA 4 City Assistant System

Schwehr, Julian; Luthardt, Stefan; Dang, Hien et al.

(2019)

DOI (TUprints): <https://doi.org/10.25534/tuprints-00014296>
License: only the rights of use according to UrhG
Publication type: Article
Division: 16 Department of Mechanical Engineering
18 Department of Electrical Engineering and Information Technology
Original source: <https://tuprints.ulb.tu-darmstadt.de/14296>

Applications

Julian Schwehr*, Stefan Luthardt, Hien Dang, Maren Henzel, Hermann Winner, Jürgen Adamy, Johannes Fürnkranz, Volker Willert, Benedikt Lattke, Maximilian Höpfl and Christoph Wannemacher

The PRORETA 4 City Assistant System

Der PRORETA 4 Stadtassistent

Adaptive maneuver assistance at urban intersections using driver behavior modeling

Adaptive Manöverunterstützung an städtischen Kreuzungen durch Fahrerverhaltensmodellierung

<https://doi.org/10.1515/auto-2019-0051>

Received April 14, 2019; accepted July 26, 2019

Abstract: The use of machine learning in driver assistance systems allows to significantly enhance their functionalities. In particular, it allows to personalize systems by evaluating the driver's past behavior. Such personalization is especially relevant for recommendations in maneuvers where the specific maneuver embodiment strongly depends on the driver's momentary driving style and attention. Led by this idea, PRORETA 4 developed a prototypical City Assistant System, which gives the driver a personalized recommendation in urban scenarios. To adapt the recommendations and warnings appropriately, the system incorporates the learned momentary driving style and the driver's gaze behavior. In this work, we describe the main functional blocks of the system, present our solutions to major implementation challenges and also discuss the safety of the used learning algorithm.

***Corresponding author: Julian Schwehr**, Control Methods and Robotics, Technische Universität Darmstadt, Darmstadt, Germany, e-mail: julian.schwehr@rmr.tu-darmstadt.de
Stefan Luthardt, Jürgen Adamy, Volker Willert, Control Methods and Robotics, Technische Universität Darmstadt, Darmstadt, Germany, e-mails: stefan.luthardt@rmr.tu-darmstadt.de, adamy@rmr.tu-darmstadt.de, vwillert@rmr.tu-darmstadt.de
Hien Dang, Johannes Fürnkranz, Knowledge Engineering, Technische Universität Darmstadt, Darmstadt, Germany, e-mails: hiendang@ke.tu-darmstadt.de, fuernkranz@ke.tu-darmstadt.de
Maren Henzel, Hermann Winner, Automotive Engineering, Technische Universität Darmstadt, Darmstadt, Germany, e-mails: henzel@fzd.tu-darmstadt.de, winner@fzd.tu-darmstadt.de
Benedikt Lattke, Continental, Frankfurt/M., Germany, e-mail: benedikt.lattke@continental.com
Maximilian Höpfl, Christoph Wannemacher, Continental, Babenhausen, Germany, e-mails: maximilian.hoepfl@continental-corporation.com, Christoph.Wannemacher@continental-corporation.com

Keywords: advanced driver assistance systems, online adaptation, driver modeling, driver behavior, intersection scenarios

Zusammenfassung: Der Einsatz von maschinellem Lernen in Fahrerassistenzsystemen ermöglicht es, deren Funktionalitäten deutlich zu verbessern. Insbesondere ermöglicht es die Personalisierung von Systemen, indem das bisherige Verhalten des Fahrers ausgewertet wird. Eine solche Personalisierung ist besonders relevant für Empfehlungen in Manövern, bei denen die spezifische Ausführung des Manövers stark vom momentanen Fahrstil und der Aufmerksamkeit des Fahrers abhängt. Ausgehend von dieser Idee entwickelte PRORETA 4 einen prototypischen Stadtassistenten, der dem Fahrer eine personalisierte Empfehlung in städtischen Kreuzungsszenarien gibt. Um die Empfehlungen und Warnungen entsprechend anzupassen, berücksichtigt das System den erlernten momentanen Fahrstil und das Blickverhalten des Fahrers. In dieser Arbeit werden die wichtigsten Funktionsblöcke des Systems beschrieben, Lösungen für die wichtigsten Herausforderungen bei der Implementierung präsentiert und auch die Sicherheit des verwendeten Lernalgorithmus diskutiert.

Schlagwörter: Fahrerassistenzsysteme, Online-Adaption, Fahrermodellierung, Fahrerverhalten, Kreuzungsszenarien

1 Introduction

1.1 Motivation

Current statistics from 2017 in Germany show that most (over 68 %) road traffic accidents involving personal injury

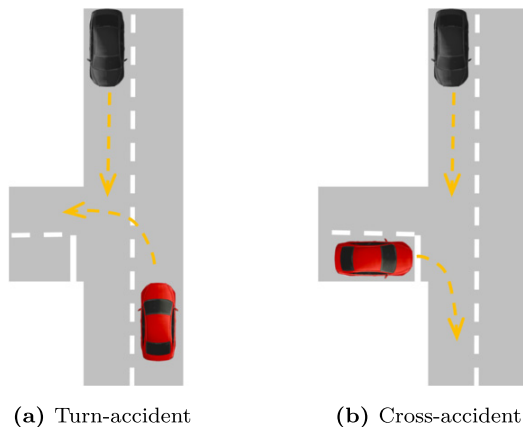


Figure 1: Risks in crossing scenarios: the ego-vehicle (red) has to give right of way to the other road users (black).

occur within urban areas. More than 60 % of the slightly injured, more than 50 % of the seriously injured and about 30 % of the killed people have been involved in an accident in this area. The main type of accidents are (with 28 %) collisions caused by turning into a road or by crossing a road (see Fig. 1). This is followed by accidents between vehicles following the roadway with 19 %, in which a collision occurs between road users moving in the same or opposite direction. The third most common type are accidents caused by turning off the road (15 %). Here a collision between a road user, who wants to turn, and a road user from the same or opposite direction at intersections, junctions or driveways (see Fig. 1a) takes place. Accidents in which the rear vehicle collides with the preceding vehicle because the front vehicle is decelerating or standing still waiting to turn count as “turn-off accident” and not “accident following the roadway” [1].

By comparing these accident statistics from 2017 with the Advanced Driver Assistance Systems (ADAS) that are currently available for urban areas, it becomes clear that there is a lack of actively supporting systems. There are many systems available from different vehicle manufacturers that support the driver in the longitudinal vehicle guidance or warn or even intervene to prevent potential front collisions [2, 3] or collisions at junctions [3, 4].

However, such systems do not provide active support in maneuvering decisions, e. g., for turning. For example, there is currently no system that tells the driver whether the current gap between two oncoming vehicles suffices to turn left, or whether the driver should wait for a larger gap. This is partly due to the fact that for active maneuver recommendations the individual driving style of the driver plays a major role. A general design of driver assistance systems, as it is done in the context of “common” collision

warning systems, is not possible here. If, for example, a very conservative recommendation is chosen in the system design for a left-turn recommendation, a sporty driver will find this patronizing and will not accept the system. A risk-averse design in turn would be considered to be too dangerous by a cautious driver. This is especially critical since a cautious driver with her driving skills may not be able to take a short or small gap between two vehicles in the oncoming traffic. Also the current attention plays an important role. Maybe the driver is distracted if she is waiting for a suitable gap in oncoming traffic and turns to a side activity like looking on her mobile phone. Such a distracted driver needs a longer time to get back into the situation and for executing the turn than a driver who constantly observes the traffic and is immediately ready to go. Therefore, there is a need for functional customization in terms of driving style and driver’s attention, in order to develop systems that actively assist the driver in making the maneuver decision.

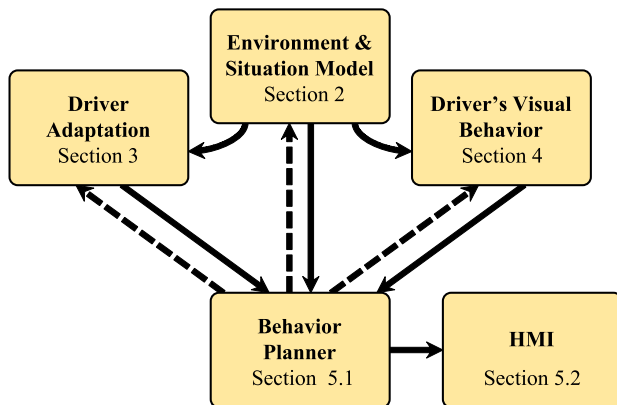
TU Darmstadt meets this demand together with Continental AG in the research project PRORETA 4. This research project was dedicated to the usage of machine learning in driver assistance systems in order to adapt them individually to the situation and the driver. The project was scheduled from 2015 to 2018 and four research assistants from three different institutes of TU Darmstadt worked together on this interdisciplinary project. Within this frame, several articles comprising new algorithms for driver intention detection and online driver adaptation [5–9], visual localization and mapping [10–13] and driver gaze target estimation [14–17] have been published as well as articles on safety approval of machine learning algorithms in the automotive context [18]. Many of the core ideas can be retrieved in the exemplary prototypical assistance system that is presented in this work.

1.2 City Assistant System – system overview

As part of the research project, the PRORETA 4 City Assistant System was developed which issues an individual, situation-adapted maneuver recommendation or warning in urban traffic situations by observing the driver, her current condition and the environment. This maneuver recommendation was implemented for three use cases on a test vehicle: turning left with oncoming traffic, entering a roundabout, and approaching and passing a left-yields-right intersection. In order to emphasize the benefit of driver adaptation, the credo of the system design is to use straightforward techniques and approaches to solve each of the encountered sub-problems but still reach a complex

Table 1: Overview on the methods used in the City Assistant System and their connection to the research fields in PRORETA 4.

Research Field	Realized Solution for the City Assistant System	Advanced Approach investigated in PRORETA 4
Localization & Mapping	Scenery Model: GNSS fused with speed and acceleration, digital HD-map, see Section 2.1	Camera-based long-term localization with LLama-SLAM, see [11, 13]
Situation Comprehension	Situation Model: geometrical assignment of road users to segments of the situation roads, lane assignment with fuzzy logic, see Section 2.2; Gap Model: geometrical representation of gaps in a situation coordinate system, rule-based “lazy” classification of relevant objects, see Section 2.3	(outside of the project’s focus)
Driver Adaption	Driving style detection: unsupervised learning using k -Means; Online adaption: concept of gap acceptance for predicting left turn, see Section 3.1	Learning the dependency between maneuver executions for personalization using LSTM [6]
Safety approval	4-step approach to identify lack of generalizability of learned algorithms, see Section 3.2	Application to extended scope (to be published)
Object-of-Fixation	Driver’s perception as simple threshold model as in [26] (but in 2D) with minimum fixation time, see Section 4	Tracking the driver’s gaze target with Bayesian Filter, see [15–17]
System coordination	Rule based approach with state machine, see Section 5.1	(outside of the project’s focus)

**Figure 2:** Overview of the function blocks that compose the PRORETA 4 City Assistant System. Each of the blocks is described in more detail in the indicated section. The solid arrows depict the flow of information to generate and deliver the adaptive recommendation to the driver. The dashed arrows indicate the situation dependent control of the function blocks, which is described in Section 5.1.1.

and intelligent system behavior through the interplay of the function blocks. These different function blocks of the PRORETA 4 City Assistant System are shown in Fig. 2 and Tab. 1 lists the approaches taken in a short overview. They firstly include an environment perception and situation comprehension module (Section 2). Secondly, a module to grasp the driver style and infer the individual adaption, is presented and analyzed under safety aspects (Section 3.1 and 3.2). This module incorporates machine learning tech-

niques in the otherwise computational intelligence-based approach of the overall system. Thirdly, the incorporation of the driver’s visual behavior is addressed (Section 4). These three main function blocks are controlled by a behavior coordination and planning module (Section 5.1) that sends out all necessary information to the *Human Machine Interface* (HMI). As part of the presentation of the HMI in Section 5.2, the individual use cases of the assistance system will be described in more detail. Section 6 concludes the key points of the project.

2 Environment model & situation comprehension

To enable the City Assistant System to give recommendations in left-turn scenarios, the system has to perceive, analyze and comprehend the oncoming traffic. In our system this task is performed in a three-step perception and comprehension pipeline which is depicted in Fig. 3. Therein each layer builds upon the preceding layer and enhances and refines the information. The first layer is called the *Scenery Model* and contains the static environment as well as the position of the ego-vehicle within this environment. The median layer is the *Situation Model* which adds the dynamic elements, i. e., the other road users, to the model and combines them with the scenery to obtain a convenient representation of the present situation. In the third and final layer, called the *Gap Model*, the information from

the *Situation Model* is converted to a more meaningful representation for the actual application, i. e., a list of gaps between relevant road users is compiled. The following paragraphs will explain these three layers of the pipeline in more detail and highlight the main problems they solve.

2.1 Scenery model

The scenery model provides information about the surrounding static environment. In the case of the City Assistant System the static environment is not perceived by the vehicle's sensors but solely employs road data from a digital map. All roads in the used HD-map are represented as a list of nodes with geographical coordinates and have a set of attributes. For our application we use the road attributes: connected roads, number of forward or backward lanes, road width, road class and speed limit. To use this map data and combine it with objects detected by the vehicle's sensors, the system also needs the position and orientation of the ego-vehicle. This data is provided by a Kalman Filter that fuses GNSS¹ readings with speed and acceleration measurements from the vehicle sensors. To improve the system's capabilities, the ego-vehicle's position could also be determined by a camera-based long-term localization, which was also a research subject in PRORETA 4 [11, 13]. All geographical coordinates inside the map as well as vehicle position and orientation are transformed from the common WGS84 coordinate system with latitude

and longitude values in an UTM²-based coordinate system. This simplifies the handling of spatial information in all following steps, since this is a flat Cartesian system and all coordinates are given in meters.

2.2 Situation model

In the second layer of the environment pipeline, other road users are added to the model and associated with the static environment. The City Assistant System uses a front-mounted long-range radar (Continental ARS) and two short-range radars (Continental SRR) at the front corners of the vehicle to detect other road users. The two short-range radars are rotated $\pm 65^\circ$ to the side to capture the crossing traffic in roundabout and left-yields-right scenarios. Both radar systems provide a preprocessed list of detected objects which specify the measured position, orientation, dimensions and velocity of each object.

The main task of this layer is to associate the objects to specific roads in the scenery model to enable the gap computation in the next layer. Therefore, the object data from the radars is firstly transformed from the moving sensor coordinate systems to the stationary coordinate system of the scenery model using the known position, orientation, speed and turning rate of the ego-vehicle. Subsequently, the algorithm determines for each object the nearest road segment and its position on this segment. Here, a road segment is the stretch between two nodes of a road in the scenery model, respectively the digital map (see Fig. 3). After associating the objects to road segments, the objects' positions are given in "road coordinates", i. e., the position of an object is specified by a road, a segment on the road and its longitudinal and lateral position on that segment (see drawings in Fig. 3). This is very useful for the following gap computation and is also similar to how human drivers think about the positions of other road users. However, simply assigning an object always to the next road does not always result in proper associations, especially in intersection scenarios. Therefore, we restrict the candidate roads for an assignment to a very limited set which depends on the current situation and is provided by the behavior planner (compare Section 5.1.1). This set is called "situation roads" and does contain only the relevant roads with the right of way in this situation, e. g., the straight

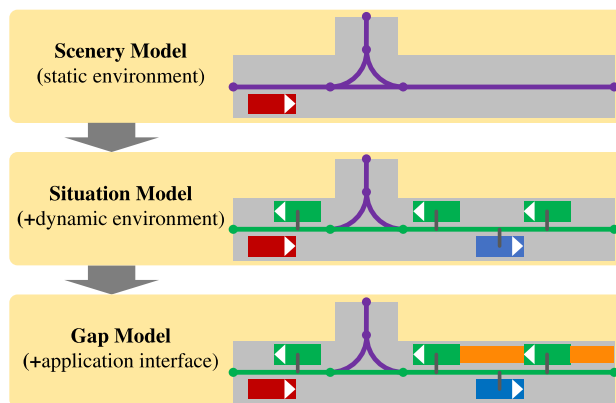


Figure 3: Environment perception and situation comprehension pipeline used in the City Assistant System. Each of the three layers enhances and refines the situation information. The ego-vehicle is depicted in red and the oncoming traffic in green. The gaps are colored orange.

¹ Global Navigation Satellite System (e. g., GPS, GLONASS, Galileo).

² The Universal Transverse Mercator (UTM) System enables the usage of planar coordinates to specify positions on the earth's surface with great accuracy. To achieve this, different coordinate projections are used for different zones of the earth [19, p. 48ff]. In our system we use UTM zone 32 which covers the major part of Germany.

Table 2: Fuzzy input and output variables of the lane assignment algorithm.

	fuzzy variable	linguistic values
inputs	road setup	one lane forward, one lane backward, two lanes
	object orientation	in road direction, in opposite road direction
	object speed	slow, fast
	lateral position	left lane, right lane, beside the road
output	lane	definitely forward, probably forward, definitely backward, probably backward, not on road

ahead road in a left-turn situation or the circle lane in a roundabout. With this procedure we always have proper assignments for our use case and also perform a preselection of relevant objects.

Lane assignment algorithm

Besides the geometric object assignment described above, we furthermore need to know in which lane an object is driving. Is it driving on the forward lane, the backward lane³ or is it not driving on a lane of the road at all (e. g., parking)? This question is answered by a lane assignment algorithm which uses fuzzy logic [20] to combine several indications of the used lane.

Tab. 2 lists the fuzzy variables which are used by this algorithm. To determine the used lane, we incorporate the road setup, the orientation of the object, its speed, and its lateral position with respect to the road. These input variables are fuzzified with predefined trapezoid-shaped fuzzy sets for each linguistic value. An object speed of 15 km/h for example would result in a fuzzy assignment for the speed with 66.6 % *slow* and 33.3 % *fast*.

The fuzzy input variables are then interpreted according to a set of 28 rules. Listing all these rules would exceed the scope of this paper, but here are two exemplary rules to illustrate the concept:

- IF *road setup* is *one lane backward*
AND *orientation* is *in opposite road direction*
AND *speed* is *fast*
AND *lateral position* is *left or right*
THEN *lane* is *definitely backward*.
- IF *road setup* is *two lanes*

AND *speed* is *fast*
AND *orientation* is *in road direction*
AND *lateral position* is *beside the road*
THEN *lane* is *probably forward*.

The result of the second rule is *probably forward* since in our setup the orientation measurement is much more trustworthy than the lateral position for fast object speeds. Therefore, we focus in such cases on the object orientation. To compute the output values of the fuzzy system, inference is performed using the 28 rules with min and max as fuzzy operators. The result of the inference is a fuzzy distribution for the variable *lane*.

In the defuzzification step this fuzzy distribution is first transformed into the three weights

$$w_F = 1 \cdot \mu_{\text{definitely forward}} + 0.75 \cdot \mu_{\text{probably forward}} + 0.25 \cdot \mu_{\text{probably backward}},$$

$$w_B = 1 \cdot \mu_{\text{definitely backward}} + 0.75 \cdot \mu_{\text{probably backward}} + 0.25 \cdot \mu_{\text{probably forward}},$$

$$w_N = 1 \cdot \mu_{\text{not on road}} + 0.25 \cdot \mu_{\text{probably forward}} + 0.25 \cdot \mu_{\text{probably backward}},$$

using the membership degrees μ and afterwards normalized to get

$$p_F = \frac{w_F}{w_F + w_B + w_N}, \quad p_B = \frac{w_B}{w_F + w_B + w_N},$$

and $p_N = \frac{w_N}{w_F + w_B + w_N}.$

These three pseudo-probabilities for forward (p_F), backward (p_B) and not-on-road (p_N) are the final output of the lane assignment algorithm and provide the main criteria for the object relevance classification in our system.

In conclusion, the Situation Model contains the static environment represented as roads, a list of objects associated to segments on this roads and a pseudo-probabilistic lane assignment for each object. Although this Situation Model was mainly developed for the computation of gaps in oncoming traffic, the information provided by this layer could also be used for other assistant functions, e. g., Adaptive Cruise Control or Emergency Braking.

2.3 Gap model

The Gap Model, which is the last layer, converts the information provided by the Situation Model to a data representation specifically tailored to our application. For the City Assistant System this is a list of gaps in the oncoming traffic. Fig. 4a depicts the used definition of the gaps and

³ In this context the terms “forward” and “backward” are defined in relation to the road direction which is given by the order of the road’s nodes in the digital map.

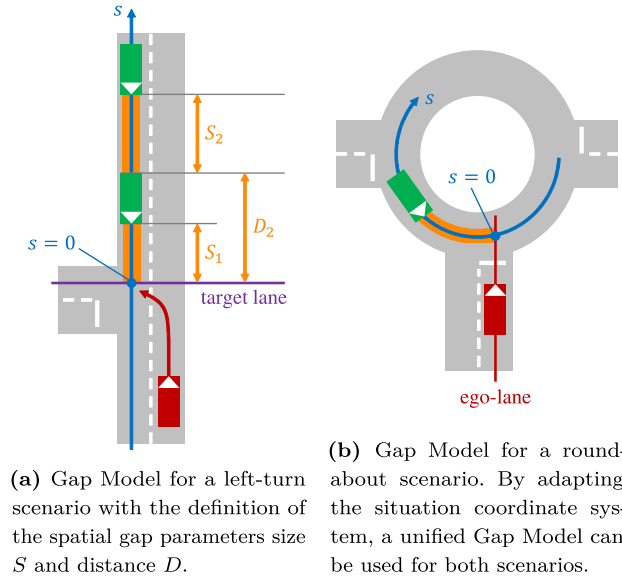


Figure 4: Specification of the used Gap Model. The gap parameters are computed in the situation coordinate system with the path coordinate s .

their associated parameters. The size S of a gap is the spatial extent of a gap starting from the rear of the leading car or the target point at $s = 0$ and ending at the front of the following car. The distance D of a gap is the distance between the *reference line* and the starting point of the gap. As reference line we choose the middle of the *target lane*, i. e., the lane on which the ego-vehicle wants to turn in. Additionally, the temporal size T and the lag or “waiting time” L of the gap are computed

$$T = \frac{S}{v_{\text{following}}}, \quad L = \frac{D}{v_{\text{leading}}},$$

where $v_{\text{following}}$ is the speed of the vehicle directly before the gap and v_{leading} is the speed of the vehicle that terminates the gap.

All gap parameters are determined in a bespoke “situation coordinate system” which is a path coordinate system with one single path coordinate s (see Fig. 4). For left-turn situations the s -curve follows the middle of the opposed lane and the $s=0$ -point is the intersection of this lane with the target lane. With this formulation, our system easily generalizes to many different situations including intersections with non-straight lanes. The second benefit of this formulation is that a roundabout situation can be handled by the same algorithm as a left-turn simply by adapting the used coordinate system. For the roundabout, the path coordinate s is measured along the middle of the circle lane and the $s=0$ -point is the intersection of the circle lane and the ego-lane (see Fig. 4b). Thus the traffic in

the circle lane can be handled exactly like the opposed traffic in a left-turn situation.

To compile the list of gaps, first all actually relevant objects, i. e., road users, from the Situation Model have to be identified. An object is classified as relevant if

- it is associated with the Situation Road,
- it is behind the reference line ($s > 0$), and
- p_F (forward) has maximum probability.

The Boolean variable r_c is set to true if all these conditions are true. However, to avoid noisy classification outputs, we do not use r_c directly but use a method we call lazy classification which combines r_c values from several consecutive time steps. The result r_L of this lazy classification is defined as

$$r_{L,k} = \begin{cases} \text{true,} & \text{if } r_{c,k} \wedge r_{c,k-1} \wedge r_{c,k-2}, \\ \text{false,} & \text{if } \bar{r}_{c,k} \wedge \bar{r}_{c,k-1} \wedge \bar{r}_{c,k-2} \wedge \bar{r}_{c,k-3} \wedge \bar{r}_{c,k-4}, \\ r_{L,k-1}, & \text{otherwise,} \end{cases}$$

where k is the index of the time step. With this method, an object only becomes relevant if all of the above relevance conditions are fulfilled for three consecutive time steps. Conversely, an object is switched to non-relevant if the conditions are missed for five consecutive time steps. We found that this simple measure considerably reduces floundering recommendations of the system caused by alternating classifications.

Before the gaps are computed, additionally, a “following ghost vehicle” is added to the list of relevant vehicles. It represents a potentially existing, yet undetected vehicle outside of the sensor range. The front of this ghost vehicle is always located at the boundary of the specified sensor range and it has an assumed default speed depending on the situation, e. g., 55 km/h on a standard inner-city road. With this trick we can also handle situations where there is no or only little oncoming traffic, without the necessity to add special rules for such cases. So an empty opposed lane would result in a gap starting at $s = 0$ and ending at the front of the ghost vehicle respectively the sensor range boundary.

By combining all these procedures inside the described three-layered perception and comprehension pipeline, we found a way to build a unified Situation Model that is completely interpretable, efficient and robust. This Situation Model respectively the refined Gap Model enables the City Assistant System to comprehend all the important entities in the current situation and their relations. This finally allows the system to give the driver reliable recommendations.

3 Driver adaptation

3.1 Driving style model

The diversity in driver behaviors is already addressed in different aspects in the literature [21]. A driver's behavior is usually conditioned by environment factors like traffic, road condition, etc. However, also in similar situations, different drivers may still behave differently. The individual behaviors are influenced by the driver's dispositional factors like driving skill, driving style, experience, emotion, etc. A general model, which does not consider the differences between drivers, leads to discomfort when using the system and lowers the trust of the driver in the system.

The individual behaviors can be observed when there is a situation where the driver can choose between different options. The left turn situation with right of way for the oncoming traffic is one of those maneuvers in which we observe the personalized decision of drivers. In a data set that we recorded from 32 drivers, the differences in preferences of choosing a gap in the left turn situation varies from about five to seven seconds. The data shows that there are some more cautious drivers who ignore even larger gaps whereas other more risk-averse drivers will always take such gaps. Similar observations on the critical personal gap are made by [22]. A driver assistant system should therefore consider the differences between drivers and provide personalized support to each individual driver.

The general approach on personalization is to collect data from the individual driver and adjust the model according to the observations. In practice, this approach requires to collect a certain amount of data in order to update the model efficiently. In the case of a left turn application, the system has to firstly collect multiple left turn maneuvers from the new driver to adjust its model accordingly. With that, the model will need a significant amount of time to adjust to the driver. If the driver changes her driving style, e. g., when she answers a phone call and shifts to a more relaxed driving style, the system needs to observe some left turn maneuvers before being able to adjust the model.

To avoid this necessity of several maneuver observations, in PRORETA 4, we exploit the correlation in behaviors of the current driver between different maneuvers to personalize the system. With the assumption that the driver's individual preferences are encoded in each maneuver execution, our approach uses past maneuvers as clues in order to personalize the prediction of the current situation (see Fig. 5). This approach allows the system to early detect intra-individual changes of the same driver

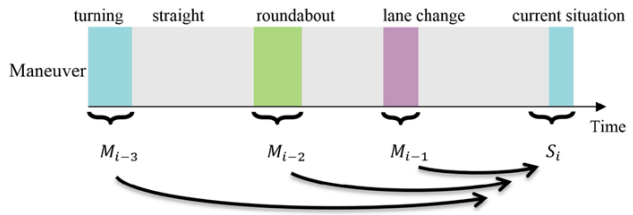
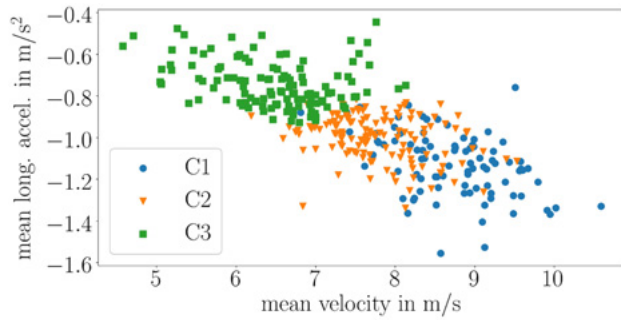


Figure 5: Incorporating past maneuver executions to personalize the prediction of current situation. M_{i-1} is the time series representing last maneuver, and S_i represent the current situation (traffic, gaps, etc.).

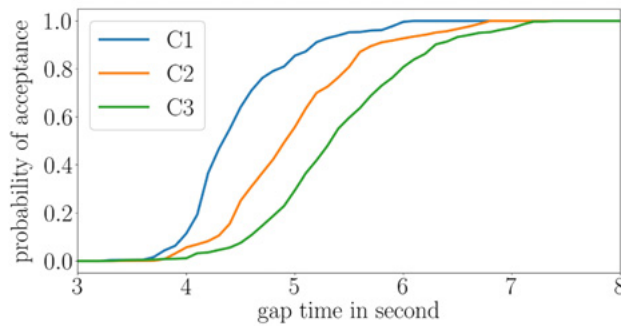
and adapt itself even when the driver has not performed any left turn maneuvers yet.

There are different ways to exploit the correlation between maneuver executions for the purpose of personalization. One way is to use supervised learning to directly learn the effect of past maneuvers on the current situation. Another approach is using unsupervised learning to extract features about the driver from previous maneuver executions. More details about the former approach can be found in [6], but in this work, we will focus on the latter one and how the adaption is realized in the vehicle. The idea of this approach is to classify a maneuver execution into different driving style groups using a clustering algorithm. For each cluster, the acceptance curve modeling the probability that a driver will take a gap of certain size is computed based on the statistics of actually taken and ignored gaps. The driver's individual acceptance curve can then be updated each time the driver executes a maneuver.

In total, we use three different maneuvers for the classification: driving through a roundabout, approaching an intersection, and turning left. Each maneuver execution is recorded as a time series of vehicle dynamics. In particular, each maneuver contains the sensor values over time of velocity, longitudinal and latitudinal acceleration, yaw rate, jerk rate and steering wheel speed. We then extract the statistical values from each signal and use them as features for clustering maneuver executions. At the training stage, we use the k -Means algorithm to separate the maneuver executions of each type into three groups. Based on the statistics of actually taken and ignored gaps, the gap acceptance is computed for each group of every maneuver type. To compute the probability of a gap being taken two assumptions are made: (a) If a driver takes a gap of size $t - \epsilon$ seconds, $\epsilon \geq 0$ then she will also take a larger gap (e. g., with size of t seconds). (b) If a driver ignores a gap with size $t + \epsilon$ then she will also ignore smaller gaps with size t . These two assumptions allow the probability of a gap with size t to be computed even if there is no data for the exact



(a) Clustering the of maneuver execution. Each dot represent a maneuver execution. The color shows the cluster assignment.



(b) The corresponding acceptance curve for each cluster computed based on the taken and ignored gaps.

Figure 6: Visualization of clustering results and the acceptance curves.

gap size of t seconds. The probability of a gap with size t being taken is

$$P(t) = \frac{\sum_{i=1}^n [g_i \leq t]}{\sum_{i=1}^n [g_i \leq t] + \sum_{j=1}^m [h_j \geq t]}$$

with n and m being the total number of taken and ignored gaps respectively and $[x]$ being the counting function that returns 1 if x is true and 0 otherwise. g_i and h_j are the size (in seconds) of taken and ignored gaps respectively.

Fig. 6a shows the clustering of the maneuver executions of approaching an intersection. The x-axis shows the mean of velocity and the y-axis shows the mean of longitudinal acceleration. Each dot is a recorded maneuver and each color shows the cluster assignment. Note that the figure is created in two-dimensional space but the actual clustering algorithm is performed in a higher dimensional space which includes further vehicle dynamic features as mentioned above. The cluster C1 contains maneuver executions with higher velocity and high longitudinal deceleration when approaching an intersection whereas C3 contains more defensive behaviors with lower velocity and lower longitudinal deceleration. C2 shows a balanced style between C1 and C3. Fig. 6b shows the cor-

responding acceptance curves for each cluster computed from the taken and ignored gaps of a left turn within the same short recording as the respective maneuver. The acceptance curve specifies the probability of accepting a gap given the gap's size, here separated for each driving style cluster. Therein, the correlation between the clustering assignment and the acceptance curve can be observed. The acceptance curve of C1 is on the left side in comparison to C2 and especially C3. This means C1 usually accepts smaller gaps than C2 and C3. On the other hand, C3 is more defensive and mostly takes larger gaps than C1.

When applying the system on the vehicle, whenever the driver executes one of the three maneuvers, her cluster assignment is computed and the corresponding acceptance curve is selected and used to gradually update her individual acceptance curve. For this, the exponentially weighted moving average (EWMA) is used to accumulate the acceptance curves computed from executed maneuvers. This update process allows the system to gradually forget the older maneuver executions and put more weight on the newer ones where the weighting parameter controls the momentariness, or respectively the persistence, of the driving style model.

3.2 Safety considerations on the driver adaptation

The fact that the learned model is significantly responsible for the subsequent recommendation of a gap in the oncoming traffic or at the roundabout leads to a high safety relevance. Although the City Assistant System is merely a recommendation and warning system, it has to be expected that the driver will rely on the recommendation given when she is getting used to it. If a driver is recommended to take a too small gap compared to her current driving style, the worst-case scenario is a collision with oncoming traffic. In order to take this fact into account, the safety of the driving style clustering was examined during the development of the system. Because of the high dimensionality of most of the learned models like neural networks, these are hard to interpret for humans and therefore regarded as black boxes. Even with the use of learned models like the here applied k -Means algorithm, which are easier to interpret due to the knowledge of how the input data is mapped to the different clusters, there is no easy way of interpreting whether the boundaries identified by the clustering reflect the reality. In [18], different methods are introduced to increase the interpretability of learned models. However, an increase in interpretability or the use

of interpretable algorithms is associated with a loss of performance of the trained model. As a result, other options were investigated to prove the safety. A critical point in this proof is the assurance of sufficient generalizability of the learned model. If the extracted relationships are based, for example, on non-representative training data, it is possible that these relationships are only valid for the training and test data, but not for input data which occur in the later operation. Regarding the driving style task, a possible fault in the training data is that there are less driving styles contained in the training data set as a result of an unstructured selection of test persons than exist in reality. Because of this imbalance in data in relation to the different clusters, the model trained on this data will produce incorrect predictions with regard to the current driving style when the non-contained driving styles occur. To counteract this problem as well as to check the presence of and avoid the causes of the lack of generalizability four consecutive steps are recommended:

1. Quality of data, tools, and procedure
2. Cause screening
3. Check on functional effects
4. Check on sensitivity

The first step is to comply with recommendations to ensure the quality of data, used tools and procedures in order to avoid causes of missing generalizability as far as possible. One advice, for example, is to analyze the training data before the actual training process with regard to their data distribution and density in order to ensure sufficient representativeness of the later operating conditions. Since, however, not all causes of lack of generalizability can be reliably avoided during the model development, the second step directly examines if several causes are existing in the learned model (e. g., overfitting). However, since this does not address all causes, the functionally correct behavior of the model is checked against requirements in the third step. This verifies the effects of lack of generalizability. There must be at least one functional requirement per feature, which describes the correct behavior of this feature in relation to the desired output. However, these requirements are limited to two or three dimensions, which is why the fulfillment of these functional requirements does not imply a functionally correct behavior and the sufficient generalizability of the entire model. For this reason, besides the functional requirements, robustness requirements are also checked which address the sensitivity of the model against various disturbances. The lack of generalizability can be proven by only applying step three and four to a learned model, but the additional use of step one and

step two beforehand decreases the necessary effort in order to remove possible causes.

Applying this approach to the *k*-Means model of the City Assistant System, it was shown that it is possible to identify incorrect behavior caused by the lack of generalizability by explicitly performing step 3 (check on functional effects): One functional requirement was that a higher steer speed during a left turn indicates a more aggressive driving style. This and other functional requirements are only fulfilled with a model configuration that uses the jerk as a feature. The same model without jerk being a feature violates this requirement even if the acceleration, from which the jerk is calculated, is contained in the feature set. Without knowing that a better result regarding the separability and the overlap of these curves is achieved when using the feature jerk, the developer would probably be satisfied with the result of the model without jerk. This would lead to a driver identification which classifies a balanced driver as a sporty one, resulting in a recommendation of too small gaps for the driver. This example shows how important the proof of functional requirements is in order to achieve a safe functional behavior.

In addition, applying step 4 (check on sensitivity) with five different robustness requirements shows for example that the robustness of the *k*-Means model against microscopic changes of the input data is high. These microscopic changes could appear in real world due to sensor noise or a different data preprocessing during the lifetime of the model (e. g., due to software updates). A change of the filtering of the training data followed by subsequent re-training of the *k*-Means model under same preconditions shows that the basic functionality of the driving style model is still maintained. The related robustness requirement does not include checking of behavior due to adversarial attacks, which is covered in step 2 of the approach.

At the end, the final clustering model, which is used in the City Assistant System, is proven safe regarding the checked functional and robustness requirements.

4 Driver visual cues for warning strategy adaptation

In order to incorporate the driver's visual behavior as one source of information into ADAS, the driver's gaze and head pose data needs to be mapped with all the other information necessary for an assistance system, e. g., ego-speed, position and motion of other road users (compare Section 2) and even traffic rules. This is a new aspect of

driver monitoring systems which are until now mainly restricted to distraction or fatigue detection. These existing system features are standalone functions which are not coupled to active assistance functions.

In PRORETA 4, the driver's visual behavior is incorporated in two ways. For the left-turn and roundabout recommendations, it is analyzed whether the driver is visually distracted and has turned to secondary tasks (e. g., texting, talking to other passengers) when waiting for a gap due to heavy oncoming traffic. In that case, the system notifies the driver when a sufficiently large gap is approaching. For the detection of distraction, an asynchronous counter is increased by 2 for each sample inside a predefined "eyes-on-road" region and decreased by 1 for each sample outside over an evaluation time span of the last 1.5 s. If the counter is negative, the driver is classified distracted. The reason for this counter is that in order to classify the driver distracted, a certain time must evolve during which the driver looks away. When the driver re-involves in the task, however, the detection latency should be as short as possible.

Secondly and more importantly, the PRORETA 4 City Assistant System monitors the driver in left-yields-right intersection scenarios where drivers usually exhibit a typical behavior consisting of slowing down and visually securing the intersection sufficiently far away from the junction [23]. The driver's approaching behavior is observed

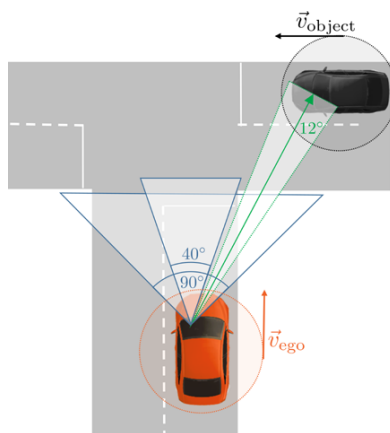


Figure 7: Incorporating driver visual cues for the adaption of warning strategies: If the driver approaches a left-yields-right intersection, it is not only checked whether the ego-speed decreases. Rather, different thresholds (here, 20° and 45° to either side) at different distances to the intersection are checked for. Additionally, if a vehicle (or other road user) with right of way approaches from the right, an early warning can be raised if, e. g., no visual fixations on the vehicle are detected and the observed TTC of the circular buffer areas is below a threshold.

and compared to this typical expected behavior. Thus, it is not only checked whether the ego-speed decreases. Rather, while approaching the intersection, it is also verified whether the driver's gaze exceeds different yaw angle thresholds at different distances to the intersection, i. e., it is checked whether the driver behaves correctly according to the situation. Additionally, if a vehicle (or another road user) with right of way approaches from the right, it is evaluated whether this object poses a potential traffic hazard and whether the driver visually perceives this object (see Fig. 7). If the risk exceeds a certain threshold, a warning would be raised in a common collision avoidance system. Here, risk is defined in terms of a simple 2D *time to collision* (TTC) with circular safety buffer [24]. Given observations of the driver's gaze, this strategy can be altered. If no fixation on the other road user is measured, the warning can be given earlier since it is assumed that the driver might have overlooked the other road user. Contrarily, if the driver has seen the other road user, an emergency brake maneuver "in the last moment" could be sufficient.

In order to bring all this information together, we built upon the scenery and situation description presented in Section 2. Traffic objects are filtered from the Situation Model so that only objects relevant for the situation remain, since for those a warning strategy is formulated. For the detection of fixations on objects in real driving setups, most models use broad tolerance thresholds for the gaze measurements for increased robustness [25–28] as remote eye tracking systems in automotive applications often do not reach the necessary precision to rely solely on the measurements themselves [17]. The opening angle of the cone around the gaze direction vector is commonly set to about 10° to 13° which corresponds to the parafovea region on the cornea of the human eye.⁴ In our system as well, a tolerance of 6° was added around the gaze yaw direction when computing the intersection with the objects' bounding boxes. Similar to the model in [26], a minimum fixation time was enforced. For increased robustness, only 90 % of the gaze samples need to intersect with the object's bounding box within the investigated fixation duration of 250 ms in order to detect a visual fixation. Further promising work in object-of-fixation detection uses tracking assumptions such that during a fixation, motion and relative geometric relations of gaze and object should be consistent [15–17].

⁴ The region of highest acuity is the fovea centralis with about 2° opening angle with the surrounding parafovea of approximately 10° [29].

A difficulty, that is out of the scope of gaze target computation, is the rapid, early and robust detection of relevant road users. In order to warn the driver, the object should be detected before the driver has seen it. In the case of a suddenly appearing object, this time frame can be as short as the duration of a typical fixation, i. e., a few hundred milliseconds. This however, is part of future research and was not tackled in PRORETA 4.

5 System coordination & HMI

5.1 Behavior planner

The Behavior Planner is the central coordination module of the City Assistant System (see Fig. 2) and has two main functions described in the following.

5.1.1 Identify the relevant situation and control the function blocks

In order to give useful recommendations, the system has to identify firstly if one of the supported use cases is currently present. To do so, the Behavior Planner analyzes the road data provided by the digital map in combination with the position of the ego-vehicle (compare Section 2.1). It determines on which road the ego-vehicle is driving and also the next junction of this road in the current driving direction. Subsequently, the upcoming junction is categorized by analyzing the connected roads and the road attributes. Four junction classes are considered in our system:

- a) intersection with left-yields-right,
- b) intersection with no right of way for the crossing road,
- c) roundabout, and
- d) other junction.

Depending on the junction class and the distance to the junction, the Behavior Planner activates and controls the three main function blocks of the system. For example, if the junction is class b) and is less than 90 m away, all blocks are activated and set-up for the left-turn scenario. Furthermore, the Behavior Planner informs the function blocks what are the relevant “situation roads” for this scenario (compare Section 2.2). The function blocks are then analyzing the situation in the background. If additionally the driver signals her intention by activating the left turn signal, the HMI is activated and starts issuing recommendations.

5.1.2 Compose the recommendation and control the HMI

The second main function of the Behavior Planner is to combine the outputs of the function blocks to a recommendation and control the HMI to issue this recommendation appropriately. For this purpose, the Behavior Planner merges the gap information from the environment and situation model (Section 2) with the gap acceptance given by the driver adaptation (Section 3.1) and labels the gaps in red and green according to their temporal size, i. e., their T -value. To stabilize these recommendations, the labels are assigned following a Schmitt trigger behavior with an additional safety buffer of 0.4 s on the lower threshold. Furthermore, the Behavior Planner computes an overall action recommendation for the driver, like “Wait for Gap”, “Prepare for Turn” and “Turn” and issues these recommendations with the right timing. Thereby it also incorporates the driver’s visual behavior, e. g., distraction (see Section 4).

The Behavior Planner also estimates the current state of the driving maneuver and incorporates that state into the recommendation. So the recommendation is frozen if the driver starts to execute the turn and is disengaged when the turn is completed. Furthermore, the Behavior Planner monitors the state of the whole system.

How the recommendation is actually visualized in the HMI, respectively issued as auditory output, is described in more detail in the following section.

5.2 HMI and recommendations

The goal of the PRORETA 4 City Assistant System is to provide the driver with individual and situation dependent recommendations and warnings tailored to their needs and capabilities. This is why also the HMI of the system, which has been developed at Continental in close collaboration with the PRORETA 4 research team, follows a multimodal, holistic approach. In addition to the visual recommendation, one can choose auditory or haptic signals as well. The general ideas of the City Assistant System, as outlined in Section 1, have been realized in three different specific use cases:

1. Give a recommendation to wait or perform a left-turn at intersections with oncoming traffic.
2. Give a recommendation to wait or enter a roundabout.
3. Give recommendations and warnings at left-yields-right intersections when the driver behavior lets to expect a violation of the way-of-right rules or does not exhibit cautious elements.



Figure 8: Recommendations for maneuver execution in the main HMI of the City Assistant System (left) with an image of the respective situation (right): The red arrow recommends to let the oncoming/crossing traffic pass. The dynamic visualization of gaps helps the driver anticipate the arrival of the “green” and thus sufficiently large gap.

5.2.1 Left-turn and roundabout scenario

For the recommendation of the gap to take, different concepts including static and dynamic visualizations had been discussed. The final choice fell on a dynamic animation of the approaching gaps in the oncoming traffic colored by the intuitive and common colors red and green signaling whether a gap is too small or large enough to take (see Fig. 8). This approach actively supports the driver to anticipate the arrival of the gap to take. It supports the intuitive and natural expectation that the first gap becomes gradually smaller before a new gap opens up. Similar dynamic display concepts were investigated in [30], supporting our design choice. A static interface in contrast, might lead to increased reaction times which would need to be considered in the system behavior by making use of a larger safety buffer. This in turn, might rather lead to a rejection instead of the acceptance of the recommendation system. In situations with dense traffic, the recommendation system relieves the driver from the stress of finding an appropriate gap suiting her needs and not to risk a too small gap. Especially inexperienced drivers could benefit from such a system.

Additionally to the visual cues depicted in Fig. 8, which are displayed in the instrument cluster, auditory signals support the driver. In cases of visual distraction, the system prompts the driver to focus on the situation when an appropriate gap arrives soon. Since roundabout scenarios exhibit high dynamics and often only short time windows of “green” gaps, here, the auditory outputs signaling the driver to enter the roundabout are of increased importance. Often, glances to the instrument cluster take too much time and a gap is gone before it can be taken. In these

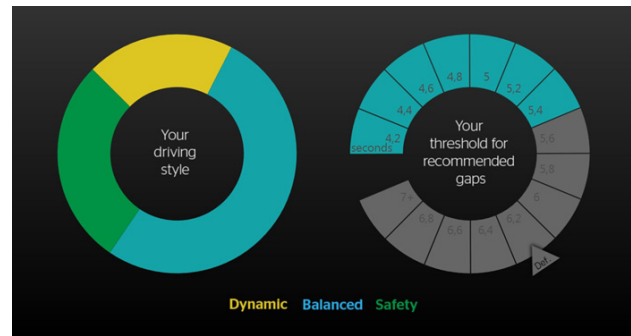


Figure 9: Driver profile display in the secondary HMI of the City Assistant System: Categorization of the current driving style on the left (legend on the bottom). The individual time gap threshold for left turn recommendations is shown on the right in the color of the predominant cluster assignment.

situations, starting to drive at the sound signal saves the necessary tenths of a second.

In order to inform the driver about her inferred driving style, additional information is provided to the driver via a personal, short-term “driver profile” on the screen in the center console (see Fig. 9).

5.2.2 Left-yields-right intersection scenario

Giving priority to the right is in many countries with right-hand driving the fundamental rule of right of way. Often, these left-yields-right intersections come without the regulation to come to a full stop, rather it is enough to slow down to be able to give way to the other road user approaching the junction from the right. Furthermore, these junctions are widely used in residential areas due to the lower speed and generally calm traffic. However, two main risk sources can be observed: Firstly, local residents are often used to their common routes and therefore know where the expectation of approaching road users is low leading sometimes to increased speed and sloppy attention. This habitual effect was also observed in our data set

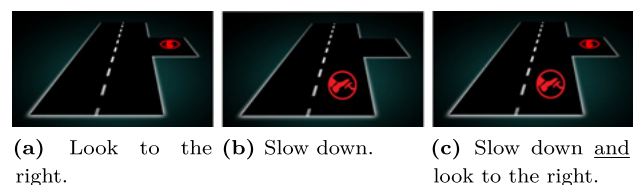


Figure 10: Warnings for left-yields-right intersections: Depending on how the driver approaches a left-yields-right intersection, they receive different HMI recommendations. Depending on the distance to the intersection, the visual cues are reinforced with an acoustical signal of increasing frequency.

where each driver drove the same route 30 times. Secondly, junctions are not necessarily well visible from afar so that unfamiliar drivers might miss these intersections. When approaching a left-yields-right intersection, drivers usually exhibit a typical behavior consisting of slowing down and visually securing the intersection sufficiently far away from the junction [23]. Building on these insights, a cascade of prompts and warnings has been constructed. If the driver does not slow down or secure visually or even fails to do both, recommendations according to the indications shown in Fig. 10 are displayed on the instrument cluster reinforced by an auditory signal with frequency increasing the closer the junction gets. When entering the intersection, it is again checked whether the driver truly secured the intersection and a feedback in form of a speech output is generated. Like all modalities (except for the visual outputs), the spoken feedback can be activated or deactivated according to the driver's needs.

6 Conclusion

Within this article, a prototypical next-generation assistance concept with a comprehensive understanding of scene, situation and driver has been presented. Within each of the presented core function blocks, straightforward approaches for individual sub-problems are implemented in the vehicle to reach the quite complex and intelligent system behavior. In the description of each building block, suggestions from the PRORETA 4 research results are given on how to continue the system's development. The concept of the PRORETA 4 City Assistant System is based on the observation that especially in complex urban scenarios, there exists no *one-fits-all*-configuration of an assistance system. By observing different meaningful maneuvers, a characteristic short-term description of the driving style is extracted with machine learning techniques which in turn is used for adaptive recommendations in other scenarios, e. g., performing a left turn or entering a roundabout. Due to this form of driving style representation, it is not necessary to perform the target maneuver (e. g., left-turn) several times, rather it is sufficient to infer the relevant information from the driving itself and provide it in a suitable representation. The safety of the resulting learned model has been analyzed to assure that the driver does not get a recommendation for too short gaps which could lead to collisions. The system is designed such that it quickly reacts to intra-individual changes of the driving style so that reasonable driver adaptation can be reached after a handful of identified maneuvers. By

constantly updating the driving style model, changes in the driving style, e. g., resulting from a suddenly occurring shortage of time, are effectively captured and the system's behavior is adapted continuously. Additionally to the driver's momentary driving style, the driver's current visual attentive state is incorporated into the situation understanding, providing insights whether the driver visually secures intersections. Based on these insights, the PRORETA 4 City Assistant System supports the driver in right of way decisions in a personalized and adaptive way. The full prototypical system presented in this article was implemented on a test vehicle provided by Continental and has been successfully demonstrated on the project's two day final event on a test route in real urban traffic.

Acknowledgment: We kindly thank Continental and all the extraordinary people working there for their great cooperation and support within PRORETA 4. Furthermore we also express our gratitude to all students that contributed to the City Assistant System. A special thanks goes to Nils Magiera, who considerably supported the PRORETA 4 project in 2017 and 2018.

Appendix

A video showing the City Assistant System in all use cases from a driver's perspective can be found at www.proreta.de. The video also presents special situations. Furthermore, an impression of the HMI can be obtained. On the project's website, additional information about the project and the final presentation event can as well be found. The project's research results beyond the City Assistant System are published in [5–18].

References

1. Statistisches Bundesamt, "Verkehr: Verkehrsunfälle 2017," ser. Fachserie 8 Reihe 7, 2018.
2. Ford-Werke GmbH. (2019) Active City Stop. Accessed Mar. 4, 2019. [Online]. Available: <https://www.ford.de/kaufberatung/informieren/technologien/sicherheit/active-city-stop>.
3. Audi AG. (2019) Fahrerassistenzsysteme. Accessed Mar. 4, 2019. [Online]. Available: <https://www.audi-mediacentr.com/de/technik-lexikon-7180/fahrerassistenzsysteme-7184>.
4. BMW Deutschland. (2016) Der neue BMW 5er. Driving Assistant Plus. Video. Accessed Mar. 4, 2019. [Online]. Available: <https://www.youtube.com/watch?v=PeilcXoPGXM>.
5. H. Dang, J. Fürnkranz, M. Höpfl and A. Biedermann, "Time-to-lane-change prediction with deep learning," in *IEEE*

- 20th International Conference on Intelligent Transportation Systems (ITSC), 2017.
6. H. Dang and J. Fürnkranz, "Using past maneuver executions for personalization of a driver model," in *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 742–748.
 7. H. Dang and J. Fürnkranz, "Exploiting maneuver dependency for personalization of a driver model," in *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen"*, ser. CEUR Workshop Proceedings, vol. 2191. CEUR-WS.org, 2018, pp. 93–97.
 8. A. Alekseenko, H. Dang, G. Bansal, J. Sanchez-Medina, C. Miyajima, T. Hirayama, K. Takeda and I. Ide, "ITS+DM Hackathon (ITSC 2017): Lane departure prediction with naturalistic driving data," *IEEE Intelligent Transportation Systems Magazine*, 2018, (Early Access).
 9. H. Dang and J. Fürnkranz, "Driver information embedding with siamese LSTM networks," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019, (Submitted).
 10. S. Luthardt, C. Han, V. Willert and M. Schreier, "Efficient graph-based V2V free space fusion," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 985–992.
 11. S. Luthardt, V. Willert and J. Adamy, "LLama-SLAM: Learning high-quality visual landmarks for long-term mapping and localization," in *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2645–2652.
 12. S. Boschenriedter, P. Hossbach, C. Linnhoff, S. Luthardt and S. Wu, "Multi-session visual roadway mapping," in *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 394–400.
 13. S. Luthardt, C. Ziegler and V. Willert, "How to match tracks of visual features for automotive long-term-SLAM," in *IEEE 22nd International Conference on Intelligent Transportation Systems (ITSC)*, 2019, (Submitted).
 14. J. Schwehr and V. Willert, "Driver's gaze prediction in dynamic automotive scenes," in *IEEE 20th International Conference on Intelligent Transportation Systems*, 2017.
 15. J. Schwehr and V. Willert, "Multi-hypothesis multi-model driver's gaze target tracking," in *IEEE 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1427–1434.
 16. J. Schwehr and V. Willert, "Tracking des Aufmerksamkeitsziels des Fahrers mittels eines Multi-Hypothesen Multi-Modell Filters," in *12. Workshop Fahrerassistenz und automatisiertes Fahren*. Uni-DAS e.V., 2018, pp. 95–105.
 17. J. Schwehr, M. Knaust and V. Willert, "How to evaluate object-of-fixation detection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019, (Submitted).
 18. M. Henzel, H. Winner and B. Lattke, "Herausforderungen in der Absicherung von Fahrerassistenzsystemen bei der Benutzung maschinell gelernter und lernender Algorithmen," in *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren*. Uni-DAS e.V., 2017, pp. 136–148.
 19. J. P. Snyder, *Map projections – A working manual*. US Government Printing Office, 1987, vol. 1395.
 20. T. Terano, K. Asai and M. Sugeno, Eds., *Applied Fuzzy Systems*. Academic Press, 2014.
 21. C. M. Martinez, M. Heucke, F.-Y. Wang, B. Gao and D. Cao, "Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2018.
 22. D. Orth, D. Kolossa, M. S. Paja, K. Schaller, A. Pech and M. Heckmann, "A maximum likelihood method for driver-specific critical-gap estimation," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 553–558.
 23. B. Metz, "Ist der Fahrer aufmerksam? Vorstellung eines Modells zur Beschreibung und Bewertung des Blickverhaltens des Fahrers," in *8. VDI-Tagung. Der Fahrer im 21. Jahrhundert: Fahrer, Fahrerunterstützung und Bedienbarkeit*, ser. VDI-Berichte, vol. 2264. VDI Verlag GmbH, 2015, pp. 187–197.
 24. J. Hou, G. F. List and X. Guo, "New algorithms for computing the time-to-collision in freeway traffic simulation models," *Computational Intelligence and Neuroscience*, vol. 2014, 2014.
 25. T. Langner, D. Seifert, B. Fischer, D. Goehring, T. Ganjineh and R. Rojas, "Traffic awareness driver assistance based on stereovision, eye-tracking, and head-up display," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3167–3173.
 26. T. Bär, D. Linke, D. Nienhüser and J. M. Zöllner, "Seen and missed traffic objects: A traffic object-specific awareness estimation," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 31–36.
 27. L. Petersson, L. Fletcher and A. Zelinsky, "A framework for driver-in-the-loop driver assistance systems," in *IEEE Intelligent Transportation Systems*, 2005.
 28. S. M. Zabihi, S. S. Beauchemin, E. A. M. de Medeiros and M. A. Bauer, "Frame-rate vehicle detection within the attentional visual area of drivers," in *IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 146–150.
 29. A. Duchowski, *Eye tracking methodology: Theory and practice*, 2nd ed. London: Springer, 2007.
 30. A.-K. Kraft, C. Maag, A. Neukum and M. Baumann, "Mensch-Maschine-Interaktion bei manuellem und automatisiertem kooperativen Fahren an Auffahrten und Kreuzungen," in *12. Workshop Fahrerassistenz und automatisiertes Fahren*. Uni-DAS e.V., 2018, pp. 56–66.

Bionotes



Julian Schwehr

Control Methods and Robotics, Technische Universität Darmstadt, Darmstadt, Germany
julian.schwehr@rmr.tu-darmstadt.de

M. Sc. Julian Schwehr is currently Software Function Coordinator at the Department Cruising Functions at Business Unit ADAS of Continental's Division Chassis & Safety in Lindau. The mentioned research on driver monitoring and driver gaze target estimation originated from his time as a research associate at the Control Methods and Robotics Laboratory of Technische Universität Darmstadt.

**Stefan Luthardt**

Control Methods and Robotics, Technische Universität Darmstadt, Darmstadt, Germany
stefan.luthardt@rmr.tu-darmstadt.de

M. Sc. Stefan Luthardt is a research associate at the Control Methods and Robotics Laboratory of Technische Universität Darmstadt. His main research focus is camera-based ego-vehicle localization, especially visual long-term SLAM. But he is also interested in traffic object tracking and worked on the modeling and the interpretation of urban traffic situations.

**Hermann Winner**

Automotive Engineering, Technische Universität Darmstadt, Darmstadt, Germany
winner@fzd.tu-darmstadt.de

Prof. Dr. rer. nat. Hermann Winner is Head of the Institute of Automotive Engineering of Technische Universität Darmstadt. He accompanied Proreta from the first generation of the project series and led the project series from the third up to the current fifth generation.

**Hien Dang**

Knowledge Engineering, Technische Universität Darmstadt, Darmstadt, Germany
hiendang@ke.tu-darmstadt.de

M. Sc. Hien Dang is a research associate at the Department of Knowledge Engineering of Technische Universität Darmstadt. His main field of activity is adaptive personalization for advanced driver assistance systems.

**Jürgen Adamy**

Control Methods and Robotics, Technische Universität Darmstadt, Darmstadt, Germany
adamy@rmr.tu-darmstadt.de

Prof. Dr.-Ing. Jürgen Adamy ist Head of the Department of Control Methods and Robotics of Technische Universität Darmstadt. Main fields of activity: Control theory, computational intelligence, autonomous mobile robots.

**Maren Henzel**

Automotive Engineering, Technische Universität Darmstadt, Darmstadt, Germany
henzel@fzd.tu-darmstadt.de

M. Sc. Maren Henzel is currently project manager of the Project group Enhanced ADAS & Tire Interactions at the Department Advanced Technology of Continental's Division Chassis & Safety in Frankfurt/M. Main fields of activity: Development of innovative driver assistance concepts and applications of the automation of driving functions. The mentioned research on safety approval of learned assistance systems originated from her time as a research associate at the Institute of Automotive Engineering of Technische Universität Darmstadt.

**Johannes Fürnkranz**

Knowledge Engineering, Technische Universität Darmstadt, Darmstadt, Germany
fuernkranz@ke.tu-darmstadt.de

Prof. Dr. Johannes Fürnkranz is Head of the Knowledge Engineering Group at Technische Universität Darmstadt. Main fields of activity: Machine Learning and Data Mining, in particular Inductive Rule Learning, Interpretable ML, Preference Learning, and Multi-label Classification, with applications in Humanities, Engineering, and Game Playing.

**Volker Willert**

Control Methods and Robotics, Technische Universität Darmstadt, Darmstadt, Germany
vwillert@rmr.tu-darmstadt.de

Dr.-Ing. Volker Willert is group leader at the Department of Control Methods and Robotics of Technische Universität Darmstadt. Main fields of activity: Pattern recognition, machine learning, computer vision, multi-agent systems, distributed optimization and control.

**Maximilian Höpfl**

Continental, Babenhausen, Germany
maximilian.hoepfl@continental-corporation.com

Dipl.-Ing. Maximilian Höpfl is Senior Expert Interior Sensing and Search Field Leader Holistic HMI and Health at the Department of Systems & Technology at Continental's Division Interior in Babenhausen. Main fields of activity: System/Cross-Discipline Engineering and Human Centered Technologies.

**Benedikt Lattke**

Continental, Frankfurt/M., Germany
benedikt.lattke@continental.com

Dr.-Ing. Benedikt Lattke is Head of Parking & Trailer Functions at Business Unit ADAS of Continental's Division Chassis & Safety in Frankfurt/M. Main Fields of activity: Development of assisted and automated driving functions for low speed maneuvering systems.

**Christoph Wannemacher**

Continental, Babenhausen, Germany
Christoph.Wannemacher@continental-corporation.com

M.A. Christoph Wannemacher is Project Manager at the Department of Systems & Technology at Continental's Division Interior in Babenhausen. Main fields of activity: Human-Machine Interaction, User Experience and Driver Model.